

curl commands cheat sheet

curl commands cheat sheet serves as a comprehensive guide for developers, system administrators, and IT professionals who frequently interact with APIs, test endpoints, or transfer data across networks. This article provides a detailed overview of the most essential curl commands, demonstrating their use cases, syntax, and options to optimize HTTP requests and responses. Whether you need to perform simple GET requests, submit forms, handle authentication, or debug network issues, mastering curl commands can significantly enhance productivity and troubleshooting efficiency. The cheat sheet also covers advanced features like handling cookies, setting headers, uploading files, and managing SSL certificates. By integrating these commands into your workflow, you can automate tasks and manage data transfers more effectively. Below is a structured outline of the main topics covered in this cheat sheet, designed to facilitate quick reference and practical application.

- Basic curl Commands
- Handling HTTP Methods
- Working with Headers and Data
- Authentication Techniques
- File Upload and Download
- Managing Cookies and Sessions
- SSL and Security Options
- Debugging and Verbose Output

Basic curl Commands

Understanding the foundational curl commands is essential for anyone working with HTTP requests. The most basic usage of curl involves fetching content from a URL using a simple GET request. This section outlines the primary commands for retrieving data and saving it to a file, as well as options for quiet operation and displaying response headers.

Simple GET Request

The simplest curl command retrieves the content of a web page or an API endpoint using the GET method. This command prints the response directly to the terminal or command prompt.

- **curl http://example.com** - Fetches the content from the specified URL.

Saving Output to a File

Instead of displaying the response, curl can save it directly to a file using the -o or -O options. The -o option allows specifying a filename, while -O saves the file with the remote name.

- **curl -o filename.html http://example.com** - Saves output to filename.html.
- **curl -O http://example.com/file.zip** - Saves using the remote file name.

Silent and Include Headers

For scripting or automation, suppressing progress meters and including HTTP headers in the output can be crucial. The -s flag silences progress, while -i includes headers.

- **curl -s http://example.com** - Runs curl silently, hiding progress.
- **curl -i http://example.com** - Includes HTTP response headers in the output.

Handling HTTP Methods

curl supports all standard HTTP methods beyond GET, such as POST, PUT, DELETE, PATCH, and HEAD. Specifying the HTTP method is critical when interacting with RESTful APIs or performing data submission and updates.

POST Requests

POST requests are commonly used to submit data to a server. curl allows sending data using the -d flag, which automatically sets the request method to POST.

- **curl -d "param1=value1¶m2=value2" http://example.com/post** - Sends form data via POST.
- **curl -X POST -d '{"key":"value"}' -H "Content-Type: application/json" http://example.com/api** - Sends JSON data with explicit POST method and content type.

PUT and DELETE Requests

To update or delete resources, curl supports the PUT and DELETE methods using the -X flag to specify the method.

- **curl -X PUT -d '{"name":"newname"}' -H "Content-Type: application/json" http://example.com/resource/1** - Updates a resource.
- **curl -X DELETE http://example.com/resource/1** - Removes a resource.

HEAD Requests

The HEAD method fetches headers only, without the response body. This is useful for checking resource metadata or HTTP status.

- **curl -I http://example.com** - Retrieves HTTP headers using the HEAD method.

Working with Headers and Data

Headers and data play a crucial role in HTTP communication. curl enables users to customize request headers, send data via forms or JSON, and manipulate content types to meet server expectations.

Setting Custom Headers

Adding or modifying HTTP headers can control caching, content negotiation, or authorization. The -H option sets custom headers.

- **curl -H "Accept: application/json" http://example.com/api** - Requests JSON response.

- **curl -H "Authorization: Bearer token_value" http://example.com/protected** - Adds an authorization token.

Sending Form Data

curl supports sending form-encoded data using the -d flag, suitable for traditional form submissions.

- **curl -d "name=John&age=30" http://example.com/form** - Sends URL-encoded data via POST.

Sending JSON Data

When working with APIs, JSON is the common data format. Setting the appropriate content-type header is necessary to inform the server of the payload format.

- **curl -X POST -H "Content-Type: application/json" -d '{"username":"john","password":"doe"}' http://example.com/login**

Authentication Techniques

Many APIs and services require authentication to protect resources. curl provides multiple methods to handle authentication including Basic, Digest, and Bearer tokens.

Basic Authentication

Basic authentication sends a username and password encoded in base64. curl simplifies this with the -u option.

- **curl -u username:password http://example.com/protected** - Accesses a protected resource using basic auth.

Bearer Token Authentication

Bearer tokens are often used with OAuth 2.0. Tokens are passed in the Authorization header.

- **curl -H "Authorization: Bearer your_token_here" http://example.com/api** - Uses bearer token for authentication.

Digest Authentication

Digest authentication provides a more secure alternative to basic auth by hashing credentials. The `--digest` flag enables this method.

- **curl --digest -u username:password http://example.com/protected**

File Upload and Download

curl is versatile for transferring files to and from servers using HTTP, FTP, and other protocols. Uploading files via POST and downloading with progress indicators are common tasks.

Uploading Files

Files can be uploaded via multipart/form-data using the `-F` option.

- **curl -F "file=@/path/to/file.jpg" http://example.com/upload** - Uploads a file to the server.

Downloading Files with Progress

Downloading files and monitoring progress can be done easily with curl. The `-O` option saves files and shows progress bars by default.

- **curl -O http://example.com/file.zip** - Downloads a file with progress display.
- **curl -# -O http://example.com/file.zip** - Shows a progress bar in a different style.

Managing Cookies and Sessions

Handling cookies with curl enables session management and persistence across multiple requests. This is useful for authenticated sessions or maintaining state.

Saving Cookies

Cookies received from the server can be saved to a file for reuse.

- `curl -c cookies.txt http://example.com` - Saves cookies to a file named cookies.txt.

Sending Cookies

Previously saved cookies can be sent with requests to maintain sessions.

- `curl -b cookies.txt http://example.com/profile` - Sends cookies from the file.

Using Cookie Strings

Cookies can also be sent as inline strings for quick testing or scripting.

- `curl -H "Cookie: sessionid=abc123; theme=dark" http://example.com`

SSL and Security Options

curl provides several options to handle SSL/TLS certificates, verify server identities, and manage insecure connections, which is essential when working with HTTPS endpoints.

Disabling SSL Verification

For testing or self-signed certificates, SSL verification can be disabled, although this reduces security.

- `curl -k https://example.com` - Ignores SSL certificate errors.

Specifying CA Certificates

Custom CA bundles can be used to validate SSL certificates against trusted authorities.

- **`curl --cacert /path/to/ca-bundle.crt https://example.com`**

Client Certificates

curl supports client-side certificate authentication by specifying certificate and key files.

- **`curl --cert client.pem --key client.key https://example.com`**

Debugging and Verbose Output

Debugging HTTP requests or network issues is facilitated by curl's verbose and trace options. These provide detailed information about the request and response headers, data sent, and received.

Verbose Mode

Verbose mode outputs detailed information about the connection and data exchange, useful for troubleshooting.

- **`curl -v http://example.com`** - Enables verbose output.

Trace Mode

Trace mode saves a detailed log of the entire communication to a file or standard output.

- **`curl --trace trace.txt http://example.com`** - Logs all data to trace.txt.
- **`curl --trace-ascii trace.txt http://example.com`** - Logs human-readable ASCII trace.

Show Only Response Headers

Sometimes only the response headers are needed to analyze server behavior or caching policies.

- `curl -I http://example.com` - Fetches headers only.

Frequently Asked Questions

What is a curl commands cheat sheet?

A curl commands cheat sheet is a quick reference guide that lists commonly used curl command options and examples to help users perform HTTP requests efficiently from the command line.

How can I use curl to make a simple GET request?

You can make a simple GET request using curl by running: `curl https://example.com`. This fetches the content of the specified URL.

How do I include headers in a curl request?

Use the `-H` option to include headers. For example: `curl -H "Authorization: Bearer token" https://api.example.com`.

How can I send data using POST with curl?

Use the `-d` option to send data with a POST request: `curl -X POST -d "key1=value1&key2=value2" https://example.com/api`.

How do I save the output of a curl request to a file?

Use the `-o` option followed by the filename: `curl https://example.com -o filename.html` saves the response to `filename.html`.

What option do I use to follow redirects automatically in curl?

Use the `-L` option to make curl follow HTTP redirects: `curl -L http://short.url`.

How can I display only the response headers with curl?

Use the `-I` or `--head` option to fetch only the headers: `curl -I https://example.com`.

How do I upload a file using curl?

Use the -F option to upload files with multipart/form-data: `curl -F "file=@path/to/file" https://example.com/upload.`

Can curl be used for authentication? If yes, how?

Yes, curl supports authentication. For basic auth, use -u username:password, e.g., `curl -u user:pass https://api.example.com.`

Additional Resources

1. *Mastering cURL: The Ultimate Command Line Tool*

This book provides a comprehensive guide to using cURL for data transfer and network testing. It covers basic to advanced commands, options, and real-world examples. Readers will learn how to automate web requests, handle authentication, and troubleshoot connectivity issues efficiently.

2. *cURL Commands Cheat Sheet: Quick Reference Guide*

Designed as a handy reference, this cheat sheet book compiles the most commonly used cURL commands and options. It's perfect for developers, sysadmins, and testers who need quick reminders and practical tips. The book also includes examples for working with HTTP, FTP, and other protocols.

3. *cURL for Developers: Practical API Interaction*

Focused on developers, this book explores using cURL to interact with APIs seamlessly. It explains how to send requests, handle responses, and debug API calls. Step-by-step tutorials help readers integrate cURL into their development workflows effectively.

4. *Networking with cURL: A Hands-On Approach*

This book delves into networking fundamentals through the lens of cURL commands. Readers gain insights into protocols like HTTP, HTTPS, FTP, and how cURL facilitates communication. It's an excellent resource for network engineers and enthusiasts looking to deepen their practical knowledge.

5. *Automating Tasks with cURL: Scripts and Solutions*

Learn how to automate repetitive network tasks using cURL scripting techniques. This book covers shell scripting integration, batch processing, and error handling. It empowers readers to build efficient automation workflows for data retrieval and server management.

6. *The Art of Debugging with cURL*

An essential guide for troubleshooting network issues using cURL's diagnostic features. The book explains options for verbose output, header inspection, and SSL debugging. It equips IT professionals and developers with tools to diagnose and resolve connectivity problems quickly.

7. *cURL Command Line Essentials*

This concise book introduces the essential cURL commands every user should know. It focuses on practical usage, including downloading files, submitting forms, and managing cookies. Ideal for beginners, it lays a strong foundation for mastering cURL.

8. *Secure Data Transfers Using cURL*

Explore how to use cURL to perform secure data transfers over the internet. The book discusses SSL/TLS configurations, certificate management, and best security practices. It's a valuable resource for anyone handling sensitive data or working in secure environments.

9. *Advanced cURL Techniques: Beyond the Basics*

Targeted at experienced users, this book covers advanced cURL features such as proxy support, multipart uploads, and custom headers. It provides in-depth explanations and use cases to maximize cURL's potential. Readers will discover powerful ways to enhance their command line toolkit.

[Curl Commands Cheat Sheet](#)

Curl Commands Cheat Sheet

Related Articles

- [current construction on i 465 in indianapolis today update](#)
- [curt discovery brake controller manual](#)
- [cummins ats wiring diagram](#)

[Back to Home](#)